Information Retrieval Project

Extended Abstract

Filipe Azevedo
filipe.p.azevedo@tecnico.ulisboa.pt
82468

Martim Zanatti
martim.zanatti@tecnico.ulisboa.pt
82517

Ana Nogueira
ana.b.nogueira@tecnico.ulisboa.pt
82433

ABSTRACT

This project implements an Information Retrieval system and a classification mechanism for the analysis of political manifestos.

1 INTRODUCTION

For a given a dataset of documents containing large electoral manifestos, this project implements an *ad hoc* search solution for querying the presented corpus according to user-inputted information, subsequently presenting the obtained results ordered by their relevance. In addition, an implementation is given for the classification of documents - for a given text, the program tries to predict which political party is most likely to have produced the selected manifesto, while also computing its precision, recall, F1 and support values. At last, the document is target to statistical analysis of contents, evaluating the usage of named entities found in the documents.

2 AD-HOC SEARCH

Our initial approach to implementing *ad hoc* search was by using the **Term Weight - Inverse Document Frequency (TF-IDF)** using the *sklearn*[1] library. However we soon realized that we could achieve better results by using a more complex algorithm that also implemented the **Term Weight - Inverse Document Frequency (TF-IDF)**, so we chose to use the **Okapi BM25** [2] python library.

For this part of the project we chose to gather all the separated manifestos parts since we think that by having more information in each manifesto, the **Term Weight - Inverse Document Frequency (TF-IDF)** weights would be more accurate, and so would the results of the search be.

We decided to use the manifestos in portuguese simply because of our fascination for the language. Before trying anything, we remove all the punctuation and all the stopwords from the documents and the query. We then create an object of the BM25 type and calculate the average **Inverse Document Frequency (IDF)** for every word in the corpus. After this, we just need to calculate the BM25 scores by simply calling the `get_scores`[3] function with the refined query and the average IDF as parameters. With the most relevant documents in our hands, we decided to print some relevant information about the query: the most relevant manifestos by decreasing BM25 score, for each party how many manifestos are in the relevant documents set, how many times each party mentions each keyword in the query, the number of keywords in the query that are present in the manifestos of each party, and the total amount of query keywords that appeared in each party's manifestos (a raw count).

3 CLASSIFICATION OF DOCUMENTS

For the classification part of the project we once again chose to use the portuguese corpus. This time, however, we chose to not aggregate the separated parts of the same manifesto into one since by doing that we would have a lot less data to train the classifiers with (less instances of labeled data). This time we use a *Panda*[4] library frame to store the contents of our corpus, divided by columns that correspond to the text, manifesto_id, party, date, and title.

The section of classification was divided into two parts. A first part where we test the effectiveness of our predictions, and a second part where we classify a given query. Since we are classifying strings, we first needed to calculate the **Term Weight - Inverse Document Frequency (TF-IDF)** weights and fit them to the classifier, before its training and the prediction of labels. We calculated the weights for 1 through 3-grams because we figured that was a good middle ground: more than that and the results wouldn't change much while making the code slower and more complex.

3.1 Effectiveness of the Classification

To be able to do this we divided our corpus into two sets: a train set, consisting of about 80% of the total corpus, and a test set, consisting of the remaining 20%. We decided to use 5 different classifiers and compare the results between them. The 5 different classifiers we used are all present in the *sklearn* library: a **Naive-Bayes** classifier (`MultinomialNB` function), a **KNearest-Neighbors** classifier (`KNeighborsClassifier` function), a **Linear Model** classifier (`Perceptron` function), a **Neural Network** classifier (`MLPClassifier` function), and a **Support Vector Machine** classifier (`LinearSVC` function). Despite implementing a **Neural Network** classifier we weren't able to draw any conclusions pertaining to that classifier due to poor time management on our part (however, we still left the code in). We used every classifier with the default arguments except the **Neural Network** one where we defined the solver to be "lbfgs", the max iterations to be 10 and the number of hidden layers to be 10 as well, to limit the amount of time needed to run that classifier. After training the classifier with the training set, we predicted the labels for each document of the test set and compared the results with the actual labels of the test set. To calculate the efficiency metrics we used the `accuracy_score` and the `classification_report` functions with both the correct and the classifier predicted labels of our test set.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[2] https://radimrehurek.com/gensim/index.html

[3] https://github.com/RaRe-Technologies/gensim/blob/develop/gensim/summarization/bm25.py

[4] https://pandas.pydata.org/

3.2 Classification of a Given Query

After testing the efficiency of the 5 chosen classifiers, we decided to use the one with the most precision to classify our given query: **SVM** . Since we didn't need to test our classifier, we were able to use the entire corpus as a training set. Finally, after training the classifier, we predicted the label of the given query, or the party that the classifier predicted to written that sentence.

4 STATISTICAL ANALYSIS

We then proceeded to calculate some statistics on the content of the documents. We do this by leveraging the *spacy.io*[5] library for natural language processing. For this specific task, we used the English language data, instead of the Portuguese one, as with the Portuguese manifestos we could not get an extensive list of mentioned named entities.

We begin by processing the data and loading the language model we would use (en_core_web_sm). By running spacy.io over the data, we can determine a label for each term, which allows us to filter out the terms that do not comply with our needs. For this particular statistical exercise, we have no interest in maintaining information about times or dates, for instance.

With the usage of Panda Dataframes, we load our information from the csv file and then group the dataframe by manifesto's ID and concatenate the texts. Now we can begin to iterate over the resulting dataframe and extract the chosen statistics.

Our implementation first outputs party-specific statistics, given that each of those parties have authored at least one of the manifestos. Firstly, it presents a list with the three most commonly named entities by the current party being observed. Next, it presents the number of times the current party has mentioned each of the other parties in their manifestos.

In addition to the above, we present some global statistics for the data analysed. We decided to indicate which were the entities most mentioned by all parties, thus showing the most relevant terms in the manifesto collection. Lastly, we wanted to determine which party was the most mentioned in every other party's manifestos.

5 RESULTS

In this section we will present our results for the different functionalities of our program.

5.1 Search Solution Results

With the use of the **Okapi BM25** algorithm we were able to return the most relevant documents most times. Some successful queries include those terms that have a high correlation with a given party or manifesto. For example, the word "PCP" is highly used by the Portuguese Communist Party, which means that the algorithm will return the manifestos of that party with higher scores, Figure 1. On the other hand, examples of unsuccessful queries are those where the keywords are vague and can appear in the political context of any party. For example, the query "Partido de direita", Figure 2.

5.2 Classification Solution Results

Here we present the results of the precision, accuracy, f1, support and recall for the chosen classifiers (except **Neural Networks**) in Figure 3.

Since **Support Vector Machine** is the best classifier metric wise, we decided to show the confusion matrix for it. Figure 4 shows a confusion matrix where the rows are the correct labels and the columns are the predicted labels.

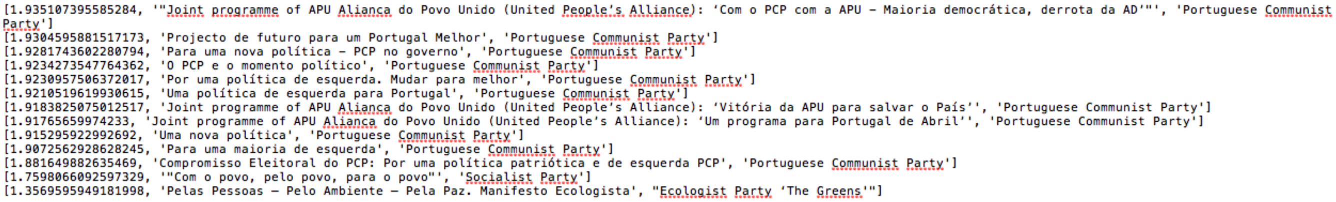
5.3 Statistics Solution Results

With our statistics solution we have concluded that the party most mentioned in other party's manifestos was the Labour Party, and we have been able to extract the most mentioned entities throughout the documents.

6 CONCLUSIONS

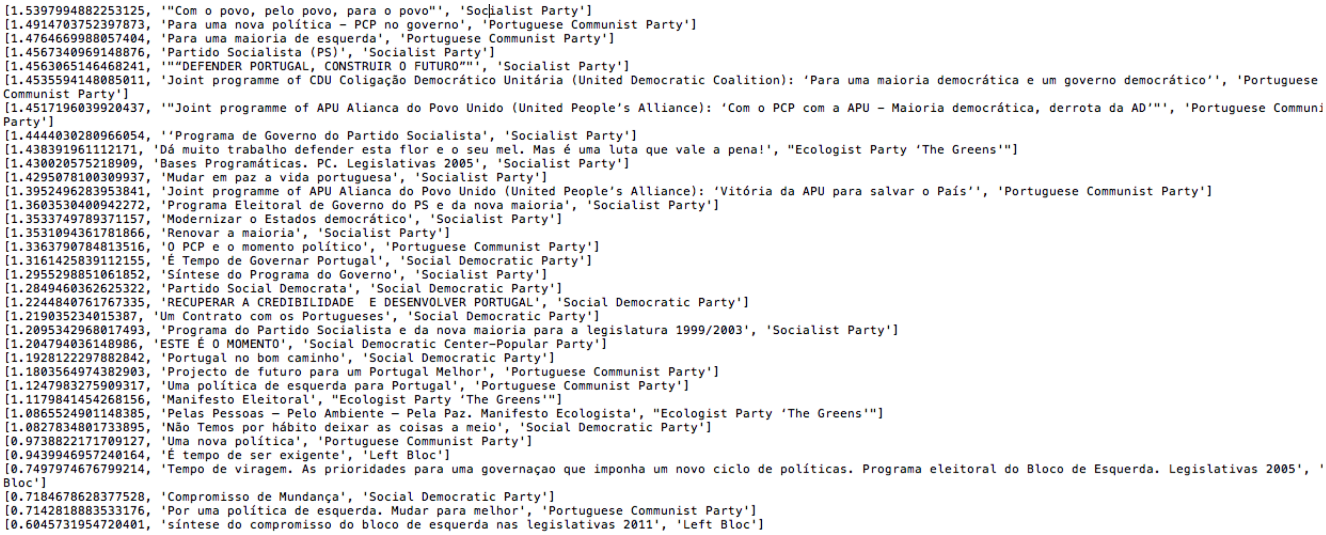
Overall, we think that this project helped us understand and learn more about Information Processing and Retrieval. In the end though, we weren't able to fully explore any of the functionalities of the project due to poor time management, but despite that, the functionalities and techniques that we did implement contributed to our better understanding of this problem.

[5] https://spacy.io/



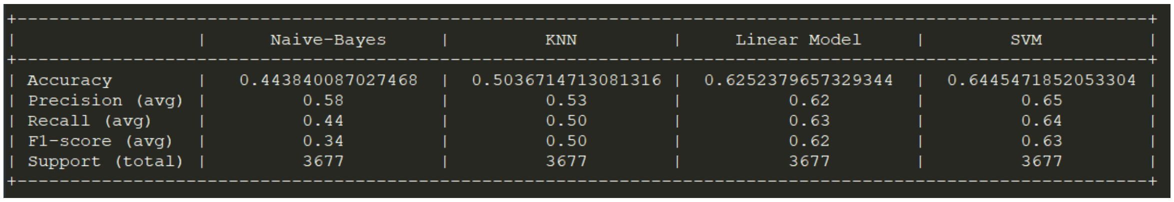
```
[1.935107395585284, '"Joint programme of APU Alianca do Povo Unido (United People's Alliance): 'Com o PCP com a APU – Maioria democrática, derrota da AD'"', 'Portuguese Communist
Party']
[1.9304595881517173, 'Projecto de futuro para um Portugal Melhor', 'Portuguese Communist Party']
[1.9281743602280794, 'Para uma nova política – PCP no governo', 'Portuguese Communist Party']
[1.9234273547764362, 'O PCP e o momento político', 'Portuguese Communist Party']
[1.9230957506372017, 'Por uma política de esquerda. Mudar para melhor', 'Portuguese Communist Party']
[1.9210519619930615, 'Uma política de esquerda para Portugal', 'Portuguese Communist Party']
[1.9183825075012517, 'Joint programme of APU Alianca do Povo Unido (United People's Alliance): 'Vitória da APU para salvar o País'', 'Portuguese Communist Party']
[1.91765659974233, 'Joint programme of APU Alianca do Povo Unido (United People's Alliance): 'Um programa para Portugal de Abril'', 'Portuguese Communist Party']
[1.915295922992692, 'Uma nova política', 'Portuguese Communist Party']
[1.9072562928628245, 'Para uma maioria de esquerda', 'Portuguese Communist Party']
[1.881649882635469, 'Compromisso Eleitoral do PCP: Por uma política patriótica e de esquerda PCP', 'Portuguese Communist Party']
[1.7598066092597329, '"Com o povo, pelo povo, para o povo"', 'Socialist Party']
[1.3569595949181998, 'Pelas Pessoas – Pelo Ambiente – Pela Paz. Manifesto Ecologista', "Ecologist Party 'The Greens'"]
```

Figure 1: Query 'PCP'



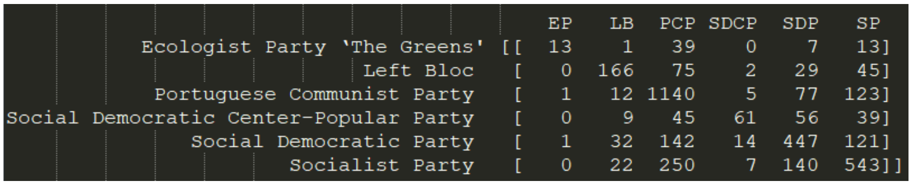
```
[1.5397994882253125, '"Com o povo, pelo povo, para o povo"', 'Socialist Party']
[1.4914703752397873, 'Para uma nova política – PCP no governo', 'Portuguese Communist Party']
[1.4764669988057404, 'Para uma maioria de esquerda', 'Portuguese Communist Party']
[1.4567340969148876, 'Partido Socialista (PS)', 'Socialist Party']
[1.4563065146468241, '""DEFENDER PORTUGAL, CONSTRUIR O FUTURO""', 'Socialist Party']
[1.4535594148085011, 'Joint programme of CDU Coligação Democrático Unitária (United Democratic Coalition): 'Para uma maioria democrática e um governo democrático'', 'Portuguese
Communist Party']
[1.4517196039920437, '"Joint programme of APU Alianca do Povo Unido (United People's Alliance): 'Com o PCP com a APU – Maioria democrática, derrota da AD'"', 'Portuguese Communi
Party']
[1.4444030280966054, ''Programa de Governo do Partido Socialista', 'Socialist Party']
[1.438391961112171, 'Dá muito trabalho defender esta flor e o seu mel. Mas é uma luta que vale a pena!', "Ecologist Party 'The Greens'"]
[1.430020575218909, 'Bases Programáticas. PC. Legislativas 2005', 'Socialist Party']
[1.4295078100309937, 'Mudar em paz a vida portuguesa', 'Socialist Party']
[1.3952496283953841, 'Joint programme of APU Alianca do Povo Unido (United People's Alliance): 'Vitória da APU para salvar o País'', 'Portuguese Communist Party']
[1.3603530400942272, 'Programa Eleitoral de Governo do PS e da nova maioria', 'Socialist Party']
[1.3533749789371157, 'Modernizar o Estados democrático', 'Socialist Party']
[1.3531094361781866, 'Renovar a maioria', 'Socialist Party']
[1.3363790784813516, 'O PCP e o momento político', 'Portuguese Communist Party']
[1.3161425839112155, 'É Tempo de Governar Portugal', 'Social Democratic Party']
[1.2955298851061852, 'Síntese do Programa do Governo', 'Socialist Party']
[1.2849460362625322, 'Partido Social Democrata', 'Social Democratic Party']
[1.2244840761767335, 'RECUPERAR A CREDIBILIDADE  E DESENVOLVER PORTUGAL', 'Social Democratic Party']
[1.219035234015387, 'Um Contrato com os Portugueses', 'Social Democratic Party']
[1.2095342968017493, 'Programa do Partido Socialista e da nova maioria para a legislatura 1999/2003', 'Socialist Party']
[1.204794036148986, 'ESTE É O MOMENTO', 'Social Democratic Center-Popular Party']
[1.1928122297882842, 'Portugal no bom caminho', 'Social Democratic Party']
[1.1803564974382903, 'Projecto de futuro para um Portugal Melhor', 'Portuguese Communist Party']
[1.1247983275909317, 'Uma política de esquerda para Portugal', 'Portuguese Communist Party']
[1.1179841454268156, 'Manifesto Eleitoral', "Ecologist Party 'The Greens'"]
[1.0865524901148385, 'Pelas Pessoas – Pelo Ambiente – Pela Paz. Manifesto Ecologista', "Ecologist Party 'The Greens'"]
[1.0827834801733895, 'Não Temos por hábito deixar as coisas a meio', 'Social Democratic Party']
[0.9738822171709127, 'Uma nova política', 'Portuguese Communist Party']
[0.9439946957240164, 'É tempo de ser exigente', 'Left Bloc']
[0.7497974676799214, 'Tempo de viragem. As prioridades para uma governaçao que imponha um novo ciclo de políticas. Programa eleitoral do Bloco de Esquerda. Legislativas 2005', '
Bloc']
[0.7184678628377528, 'Compromisso de Mundança', 'Social Democratic Party']
[0.7142818883533176, 'Por uma política de esquerda. Mudar para melhor', 'Portuguese Communist Party']
[0.6045731954720401, 'síntese do compromisso do bloco de esquerda nas legislativas 2011', 'Left Bloc']
```

Figure 2: Query 'Partido de direita'



```
+------------------------------------------------------------------------------------------------+
|                   |    Naive-Bayes      |        KNN         |    Linear Model     |     SVM          |
+------------------------------------------------------------------------------------------------+
| Accuracy          |  0.443840087027468  | 0.5036714713081316 | 0.6252379657329344  | 0.6445471852053304 |
| Precision (avg)   |       0.58          |       0.53         |       0.62          |       0.65         |
| Recall (avg)      |       0.44          |       0.50         |       0.63          |       0.64         |
| F1-score (avg)    |       0.34          |       0.50         |       0.62          |       0.63         |
| Support (total)   |       3677          |       3677         |       3677          |       3677         |
+------------------------------------------------------------------------------------------------+
```

Figure 3: Precision Metrics



```
                                          EP   LB  PCP SDCP  SDP   SP
             Ecologist Party 'The Greens' [[  13    1   39    0    7   13]
                                Left Bloc  [   0  166   75    2   29   45]
               Portuguese Communist Party  [   1   12 1140    5   77  123]
  Social Democratic Center-Popular Party  [   0    9   45   61   56   39]
                  Social Democratic Party  [   1   32  142   14  447  121]
                          Socialist Party  [   0   22  250    7  140  543]]
```

Figure 4: Confusion Matrix for SVM classifier